



成	
绩	

廣東工業大學

课 程 设 计

课程名称 数据库系统课程设计

题目名称 药品管理系统

学生学院 计算机学院

专业班级 21 人工智能 1 班

学 号 3121005358

学生姓名 欧炜标

指导教师 明俊峰

2023 年 12 月

类别	评价细则					分数	成绩
课程设计报告	要求论文（设计报告）结构，内容完整、阐述清晰正确，格式规范。论文一般包含： <ol style="list-style-type: none"> 绪论：选题背景及意义；研究现状、研究目标及技术方案；论文的结构及主要工作等。要求选题具有一定的新颖性。 可行性分析。 需求分析（功能需求、非功能需求）。 系统的概要设计、详细设计。 数据库的设计与实施： <ol style="list-style-type: none"> 概念结构设计； 逻辑结构设计； 物理结构设计； 使用数据库辅助设计工具，正确完成上述各阶段的设计及建模。 系统的编程实现。 系统测试（非必须）。 论文各部分完整，有摘要、目录、参考文献、致谢等。 必要的系统使用说明、系统安装配置说明（非必须）。 					30%	
系统的设计与实现	<ol style="list-style-type: none"> 正确、完整的阐述概要设计、详细设计、系统实现等各阶段自己的主要工作。 实现的系统能正确运行、主要功能实现、界面友好。 功能完整： <ol style="list-style-type: none"> 围绕系统核心业务，实现系统的主要功能； 通过实现的系统可对数据库中数据进行增、删、改、查操作； 有数据统计、报表等功能； 工作量充分； 选用技术恰当、结构合理、具备一定的难度和新颖性。 					30%	
数据库设计	<ol style="list-style-type: none"> 数据库概念结构设计：E-R 图正确、规范。 数据库逻辑结构设计：运用规范化理论设计及优化关系模式。 数据库物理结构设计：表结构设计正确、合理。 用到特殊的数据库技术，如：数据库辅助设计工具，模糊查询、分组查询、触发器、游标、存储过程等； 数据的完整性：正确使用实体完整性，参照完整性和用户定义的完整性等。 数据的安全性：用到用户标识与鉴别，存取控制机制，数据加密等。 					40%	
总评	<input type="checkbox"/> 优秀	<input type="checkbox"/> 良好	<input type="checkbox"/> 中等	<input type="checkbox"/> 及格	<input type="checkbox"/> 不及格	总分	

摘 要

随着计算机技术的飞速发展和经济的发展，社会的进步，计算机越来越深入到我们日常的工作学习及生活中，成为我们日常生活中不可缺少的辅助工具。医院药品管理系统以医院医疗用药管理为主线，正在实现全面的信息化。

本文以药店管理系统的开发设计为主要依据，结合学过的面向对象的可视化语言，总结前人工作经验开发设计的系统。此系统以 Eclipse 作为前台开发工具，以 MySQL 数据库做后台数据库，并使用 Navicat 和 power designer 进行数据库设计维护。本系统采用 Spring[3]+Spring MVC+MyBatis[2]（SSM）框架，以及 JSP 前端技术来开发一个性能优越、可扩展性强和安全可靠稳定的药品管理系统。系统主要实现了供货商管理、销售人员管理、药品管理、医院管理、药品进出流向管理、回款管理、销售管理、个人中心等功能模块。整个系统用简洁明快的界面展现出来，操作简单可行，用户使用方便。通过对各功能模块的测试，测试结果良好，具有一定的推广价值。

关键词：药品管理； MySQL； SSM； JSP.

目录

1	绪论	1
1.1	选题背景及意义	1
1.2	国内研究状况	1
1.3	研究方法、技术路线、开发环境	2
1.4	本文的结构及主要工作	3
2	相关技术介绍	3
2.1	Eclipse	3
2.2	Tomcat	4
2.3	JSP	4
2.4	Mysql	5
2.5	SSM 架构	5
2.5.1	Spring	5
2.5.2	Spring MVC	6
2.5.3	MyBatis	7
3	需求分析	7
3.1	系统设计目标	8
3.2	系统可行性分析	8
3.3	系统功能需求概述	8
3.4	系统设计规则	9
4	概要设计	9
4.1	架构设计	9
4.2	系统功能结构设计	11
4.3	数据库结构设计	11
4.3.1	概念结构设计	11
4.3.2	逻辑结构设计	13
4.3.3	物理结构设计	14
4.4	系统用例图	15
4.5	系统架构类图	16
4.6	事件跟踪图	17
5	详细设计	18
5.1	系统配置	18
5.1.1	应用层配置	18
5.1.2	数据库连接配置	19
5.1.3	SessionFactory 模式的配置	19
6	系统实现	20
6.1	总体说明	20
6.2	部分核心代码	22
6.2.1	前端主界面 main.JSP	22
6.2.2	控制类部分核心代码	24
6.3	功能模块实现	25

6.3.1	系统登陆实现	25
6.3.2	系统功能设计实现	25
7	系统测试	27
7.1	测试方法.....	27
7.2	测试的主要内容.....	27
8	回顾与展望	30
8.1	回顾总结.....	30
8.2	优点与不足之处.....	30
8.3	展望.....	31
参考文献		31
致 谢		32

1 绪论

1.1 选题背景及意义

药品管理系统可以为药店或医院管理者提供充足的信息和快捷的查询手段。但一直以来很多中小型药店或医院都使用传统手工的方式进行管理,这种管理方式存在着许多缺点,如效率低、数据易丢失等,容易造成药品积压甚至失效。另外时间一长,将产生大量的纸质数据,这对于查询、修改都造成很大不便。使用计算机对药品进行管理相对于手工管理而言,具有很多优点。例如:查询方便快捷、数据存储量大等。这些优点能够极大地提高了药品管理的效率,也能更好地保证药品的效期安全,是对消费者负责的态度体现。因此,开发一套适用于中小型药店或医院的药品管理系统管理软件很有必要。

我国网站大规模的信息化管理大多开始于上世纪 90 年代,要比西方发达国家晚了 10-20 年。在沿海地区和较发达的城市,高校信息化的建设近年来也发展的十分迅猛,信息技术已经广泛地被应用于高校的教学、科研以及管理等各个领域,并取得很好的效果。如清华大学和北京大学就已建成了以高速的校园网为核心,包括以学术研究、办公管理、社区服务等功能的数字化管理系统。

中国现在的医药体系还不成熟,药品管理系统也不成熟。在国内有很多软件公司开发药品管理系统,基本采用进销存模式,系统功能比较完善,但在安全、简捷、性能以及人性化设计方面还存在不足。药店的正常运转货源要得到保证,这就必须有着良好的进货入库管理功能,同时为了药品质量的保证,以及在未知药品销量时可以保证药店不至于积存大量的药品,对一些积存的药品进行合理的退货。作为药店盈利机构,药店的销售是非常重要的,同时要求透明化,药店的入库单、销售单、退货信息都必须合理的记录统计。药品的存货是药店进行正常运行的保证同时合理的库存量更好的应对某些药品需求量的变化,同时记录药品出库信息可以更好的反应药品的销售信息。在仓库中药品的储藏出现的损失,通过记录的信息可以精准核对药品量的信息。同时在仓库储存中应设置药品的库存下限,以便及时反映药品的信息,为药品进货提供可靠依据。

1.2 国内研究状况

在国外,很多药品管理系统只是医院信息管理系统的子模块,在功能上基本都可以

实现药品采购管理、药品销售管理、药品库存信息管理、药品收支管理等功能，有些药品管理模块还能实现药品信息智能报警、药品流向管理以及药品智能采购[1]等功能。就药品管理系统本身的开发语言来讲，主要是以 Java 为主，在数据库的选择方面主要是以 Oracle 数据库为主。除此之外，药品的监督机制方面健全，在最大程度上保证了人们的用药安全。

国内的药品管理系统起步较晚，这与国内计算机技术起步较晚有着直接的关系。骆鑫[4]依靠郑州大学第一附属医院的相关工作模式而开展研发的医院信息化药品管理系统，采用 PowerBuilder9.0、SQL server 数据库等技术，实现了药品管理、库存管理等多项功能。张振[5]使用 J2EE 以及 SQL serve2008 数据库等技术完成了新医大肿瘤医院药品细腻些管理系统的开发。刘嘉富[6]等人使用 SSH 禾、MySql8.0 等技术设计了基于 B/S 架构的药品管理系统。以上系统实现了药品管理系统应具备的基本功能，但是具有界面相对不友好、药品订单功能单一以及安全措施不够等缺陷。

1.3 研究方法、技术路线、开发环境

1) 研究方法:

根据所给资料得出系统的基本的框图，划分功能模块，应用语言编程，应用 Eclipse 与 MySQL 作为数据库服务器来开发本系统。系统主要实现了供货商管理、销售人员管理、药品管理、医院管理、药品进出流向管理、回款管理、销售管理、个人中心等功能模块。

2) 技术路线:

本系统基于 B/S(Browser/server, 浏览器/服务器)三层架构体系结构，这种模式采用多种标准的协议和技术，能够在任何硬件平台和软件环境下良好的运行。本系统采用 Spring+SpringMVC+MyBatis(SSM 框架技术开发一个性能优越、可扩展性强和安全可靠稳定的药店管理管理系统。

3) 开发环境

本系统采用的开发平台为 Eclipse IDE for Enterprise Java and Web Developers - 2023-12, 对 JDK1.8 进行配置，确保 Java 能够更好的运行，Web 服务器选择 Tomcat 9.0.8 数据库选择 MySQL，数据库可视化工具选择 Navicat Premium 15，操作系统为 Windows10，浏览器选择 Chrome，Firefox，Edge 等。

1.4 本文的结构及主要工作

本文主要分为六个章节，第一部分为前言，主要介绍了课题背景、研究意义、国内外研究现状、课题的研究方法、技术路线以及本文研究的主要内容等。

第二部分是相关技术的介绍，主要介绍了 JSP 技术、Eclipse 、Tomcat、 Spring、SpringMVC、SSM 架构、MyBatis 框架和 MySQL 数据库。

第三部分为需求分析，主要介绍了项目规划、可行性分析、系统设计的特点、系统设计的总体规划、运行环境等。

第四部分是概要设计，系统架构设计和功能模块设计是主要的设计内容。

第五部分是详细设计，主要介绍了系统的详细设计过程。

第六部分是系统实现，主要讲解主要的代码配置，核心代码的实现。

第七部分为系统调试与测试，对本系统进行了功能和界面的测试。

第八部分为总结与展望，主要对本文的研究工作进行了总结以及对进一步工作进行了展望。

2 相关技术介绍

2.1 Eclipse

Eclipse 是一个源代码开放的、基于 Java 的可扩展开发平台。对于它本身而言，它只是一个框架以及一组服务，用于通过插件组件来构建开发环境。幸运的是，Eclipse 附带了一个标准的插件集，其中包括 Java 开发工具（Java Development Kit，JDK）。

大多数用户很愿意将 Eclipse 当作 Java 集成开发环境（IDE）来使用，但是 Eclipse 的目标却不仅仅于此。Eclipse 另外还包括插件开发环境（Plug-in Development Environment，PDE），这个组件主要用来针对希望扩展 Eclipse 的软件开发人员的，这是因为它允许他们构建与 Eclipse 环境无缝集成的工具。因为 Eclipse 中的每样东西都是插件，所以对于给 Eclipse 提供插件，以及给用户提供一致和统一的集成开发环境而言，所有的工具开发人员都具有同样的发挥场所。

这种平等和一致性并不仅仅限于 Java 开发工具。虽然 Eclipse 是使用 Java 语言开发的，但是它的用途并不限于 Java 语言；比如，支持诸如 C/C++、COBOL、PHP 等编程语言

的插件已经可用，或预计将会推出。Eclipse 框架还可作为与软件开发无关的其他应用程序类型的基础，例如内容管理系统。基于 Eclipse 的应用程序的一个突出例子是 IBM Rational Software Architect，它构成了 IBM Java 开发工具系列的基础。

2.2 Tomcat

Tomcat 是一个应用服务器，这是一个比较轻量的程序，它被广泛应用在中小型系统和并发访问用户的情况不像在开发和调试过程。对于一个初学者，你可以做到这一点，当一台机器好 Apache 服务器上配置的 HTML 页面访问量可以响应请求使用它。事实上，Tomcat 的 Apache 服务器是扩展的一部分，但它的操作是独立的，所以 Tomcat 正在运行，它是一个独立的程序，和 Apache 独立运作。这里的技巧是，当正确配置，对于网页，运行需要做的页面。

另外，Tomcat 和 IIS，Apache 和其他 Web 服务器，具有处理功能的 HTML 页面，而这是一个 Servlet 和 JSP 容器，独立的 Tomcat Servlet 容器是默认模式。不过，Tomcat 是不如 Apache 服务器的处理静态 HTML 的能力。

2.3 JSP

JSP (JavaServer Pages) 是一种由 Sun Microsystems 提倡的动态技术标准，由许多网站共同建立。JSP 结合了 Java 程序片段(在 Scriptlet 中)和传统 HTML 文件(*.htm, *.html)，使得在 JSP 页面中的 Java 程序片段能够执行数据库操作、重新定向网页以及发送 E-mail 等功能，实现了动态系统的构建需求。所有的程序操作都在服务器端进行，客户端浏览器执行的仅是获取的结果，这大大降低了对客户端浏览器的要求，即使浏览器不支持 Java，仍然可以访问 JSP 页面。

Servlet 是 JSP 的全称 Java 服务器页面，是一种简化的设计，将 Java 扩展嵌入 HTML 语法（以<%@ %>形式）。JSP 和 Servlet 在服务器上执行，通常返回 HTML 文本给客户端浏览器。对 JSP 页面的请求首先执行程序，然后通过 Web 服务器一起返回 HTML 代码的结果给客户端。Java 程序段用于操作数据库、重定向网页以及插入页面和动态页面的功能。

JSP 通过访问标签和在 scriptlet 中编写的代码，可以发现在服务器端应用程序中的逻辑

辑资源。JSP 将网页设计和网络逻辑显示屏进行了分离，提供了基于快速、简单的复用的 Web 应用程序开发基础。JSP 是一种动态网页技术，其主要目的是从 Servlet 的逻辑中分离 HTML 代码和 Java 代码。

在 Java 代码中，服务器在客户端请求页面后生成 HTML 页面并返回给客户端浏览器。JSP 是建立大型 Web 应用程序的基础技术，与 Java Servlet 和 JSP 一起完成。JSP 具有易用性、完全面向对象、独立于平台和强大的安全保障等 Java 技术的特点。自从推出以来，许多大型科技网站都支持 JSP 服务器，如 IBM、Oracle、BEA 等，因此迅速成为商业应用中的 JSP 服务器端语言。

2.4 Mysql

数据库分为两类，一类是关系型数据库，另一类是非关系型数据库。不同的数据库存储数据的方式都有差异，也都有自己的优缺点，在此不做具体的介绍。

其中，MySQL 拥有支持多线程的特点，这样就可以方便的利用系统现有的资源，有效提高数据存储、读写的运行速度，MySQL 还提供了 TCP/IP、ODBC 和 JDBC 等多种链接方式，实现对数据库的链接；虽说其功能还不怎么强大、规模也相对较小，但是，对于本系统来说已经足够了，MySQL 还可以进行二次开发的数据库表结构，进行空间的扩展。

采用 MySQL 来实现系统数据库的建立，不仅有利于前期对系统数据的管理、整合等处理操作，同时，还有利于软件系统后期的二次开发时，对数据库的扩展操作，对于有限级的数据处理，MySQL 可以完美的胜任。

2.5 SSM 架构

SSM 是 Spring + Spring MVC + MyBatis 的中较受欢迎的开源 Java Web 应用框架[7] 一个综合框架。

2.5.1 Spring

Spring 是一个功能强大的框架，解决了很多问题上有着共同的 J2EE 开发。并提供了一种方式来管理业务对象的编程接口，用于注射的鼓励，而不是良好的编程习惯。基于

使用控制容器倒置的 **JavaBean** 属性是基于 **Spring** 的架构。然而，这是一个完整的图像，只有一部分：弹簧是在溶液中其使用容器如成品建筑物 **IOC** 涉及各级架构的完整独特。**Spring** 提供了唯一的数据访问抽象，包括 **JDBC** 框架，方便，高效地，这样可以提高效率，减少出错的可能性。架构进行数据访问 **Spring**，**Hibernate** 和绘图解决方案整合的 **O / R Spring** 也提供了一个唯一的事务管理抽象，它可以在各种管理技术的基本交易，例如 **JTA** 或者 **JDBC** 事务提供一个一致的编程模型。**Spring** 提供了用 **Java** 编写的 **AOP** 框架标准的通用语言，它给人的 **POJOs** 提供了声明式的事务管理和其他企业 - 如果你必须 - 可以实现自己的方面。这个框架足够强大，除了 **EJB** 的复杂性，使得 **EJB** 应用程序同时享受与传统的键相关联的服务。春节前国际奥委会和 **MVC** 框架还提供了一个强大而灵活的 **Web** 集成。

2.5.2 Spring MVC

Spring MVC 是一种便捷的 **MVC** (**Model View Controller** 既模型视图控制器) 软件开发模式。作为 **Spring** 框架的一个重要组成部分，**Spring MVC** 对请求处理期间涉及的各种关注点进行了合理而完全的分离，并且明确设置了相应的角色用于建模并处理整个生命周期中的各个关注点。

Spring MVC 可以很容易的得到各种支持，从 **IoC** 容器提供的依赖注入支持，到 **Spring AOP** (**Aspect Oriented Programming** 即面向切面编程) 的支持，以及数据访问层、事务管理层的支持等。**Spring** 提供的良好的中间层支持，是之前许多 **Web** 开发框架都缺少的东西。

Spring MVC 处理 **HTTP** (**Hyper Text Transfer Protocol** 即超文本传输协议) 请求的大致过程如下：

当收到一个 **HTTP** 请求时，由 **Spring** 提供的前端控制器 **DispatcherServlet** 负责分发请求。首先借助于 **Spring** 提供的 **HandlerMapping** 定位到具体的 **Controller**，然后 **DispatcherServlet** 将请求分发给 **SpringController**。**HandlerMapping** 则是用来完成客户请求到 **Controller** 之间的映射，在 **Spring** 中，这些映射是通过 **Spring-config.xml** 文件被定义的。

SpringController 将处理来自 **DispatcherServlet** 的请求。**Spring** 的 **Controller** 与 **Spring** 的 **Action** 类似，能够接受 **HttpServletRequest** 和 **HttpServletResponse**。**Spring** 在

org.springframework.web.servlet.mvc 包中为 Controller 接口提供了若干实现类。Controller 处理客户请求，这和 SpringAction 的功能是一致的。Controller 处理完客户请求后，则返回 ModelAndView 对象给 DispatcherServlet 前端控制器。

ModelAndView 包含模型（Model）和视图（View）。HTTP 请求返回的视图逻辑名，将借助 Spring 提供的视图解析器（ViewResolver）在 Web 应用中查找 View 对象，然后，DispatcherServlet 将 View 对象渲染出的结果返回给客户。

Spring MVC 处理 HTTP 请求流程如图 2.1 所示。

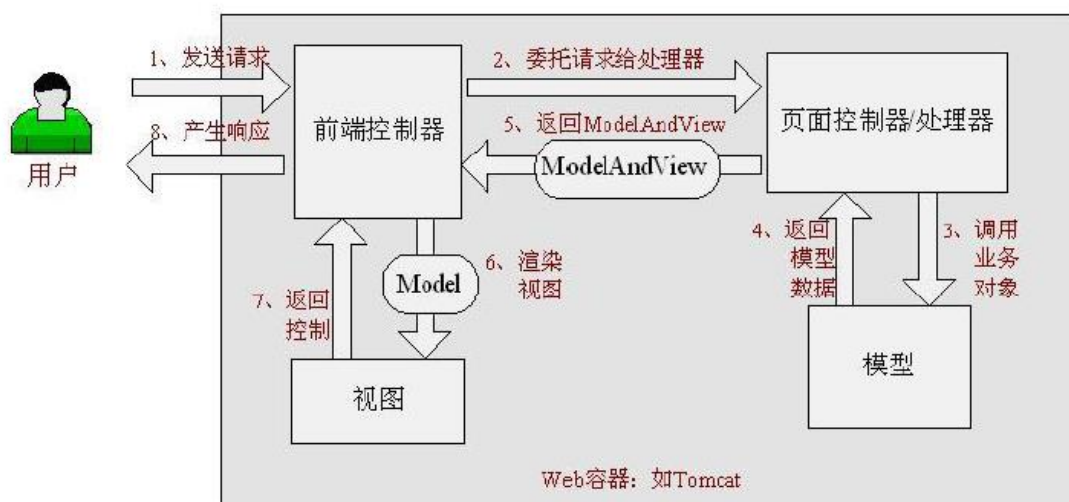


图 2.1 Spring MVC 处理 HTTP 请求流程图

2.5.3 MyBatis

MyBatis 本是 apache 的一个开源项目 iBatis, 2010 年这个项目由 apache software foundation 迁移到了 google code, 并且改名为 MyBatis。MyBatis 是一个基于 Java 的持久层框架。iBATIS 提供的持久层框架包括 SQL Maps 和 Data Access Objects (DAO) MyBatis 消除了几乎所有的 JDBC 代码和参数的手工设置以及结果集的检索。MyBatis 使用简单的 XML 或注解用于配置和原始映射, 将接口和 Java 的 POJOs (Plain Old Java Objects, 普通的 Java 对象) 映射成数据库中的记录。

3 需求分析

3.1 系统设计目标

药店管理管理系统的具体开发目标为：

系统主要实现了供货商管理、销售人员管理、药品管理、医院管理、药品进出流向管理、回款管理、销售管理、个人中心等功能模块。

3.2 系统可行性分析

1. 操作的可行性分析

系统的登录界面简单和友好，采用常见的界面窗口的登录界面，而系统的开发则采用的是较为常用的 JSP 技术。这些策使得用户不需要很长的时间就能够快速熟悉系统，并掌握系统的操作方法。此外，为了方便系统管理维护人员，用户登录系统后会给出了一些提示，使得整个系统更加人性化，用户操作更简单方便。

2. 技术可行性分析

1) 硬件可行性分析

系统的硬件要求方面不存在特殊的要求，只需要在普通的硬件配置就能够轻松的实现，只是需要确保系统的正常工作即可，以及拥有较高的效率。如果有特别低的硬件，它可以导致系统的低性能以及效率低，从而导致整个系统的运行不顺畅。以目前普遍的个人计算机的配置而言，这是十分容易实现的。因此，本系统的开发在硬件方面是可行的。

2) 软件可行性分析

Java 语言提供了一个共同的机制类似的借口动态模型，设计更集中。另外，在使用 Java 语言可以很容易实现模块化和存储信息。此外，代码复用，也可以很好的体现。因此，考虑到系统的实际情况，选择 JAVA 作为本系统开发语言的时候。通过上述分析，该系统的设计实现在软件方面是可行的。

因此，我们进行了三个方面的可行性研究，可以看出，该系统的开发是没有问题的。

3.3 系统功能需求概述

系统主要实现了供货商管理、销售人员管理、药品管理、医院管理、药品进出流向管理、回款管理、销售管理、个人中心等功能模块。

3.4 系统设计规则

自己的设计规则无论在哪个系统都要有。系统也一样，它的主要设计规则有：

简单性：对于一个系统来说是非常重要的是系统操作简单易懂的同时，也要让实现系统的功能得到实现。

针对性：该系统设计的定向开发设计是关于药店管理管理系统，专业性和针对性很强。

实用性：系统主要实现了供货商管理、销售人员管理、药品管理、医院管理、药品进出流向管理、回款管理、销售管理、个人中心等功能模块，具有良好的实用性。

4 概要设计

4.1 架构设计

系统采用 B/S 模式架构。B/S 是 Brower/Server 的缩写，客户机上只要安装一个浏览器（Browser），服务器安装 Oracle、Sybase、Informix 、SQL Server 或 MySQL 等数据库。浏览器与同数据库进行数据交互，可以通过 Web Server 实现。

在 B / S 架构中，用户通过浏览器访问时，服务器收到请求，将处理结果返回给浏览器。客户端工作进行了简化，系统三层架构如图 4.1 所示。

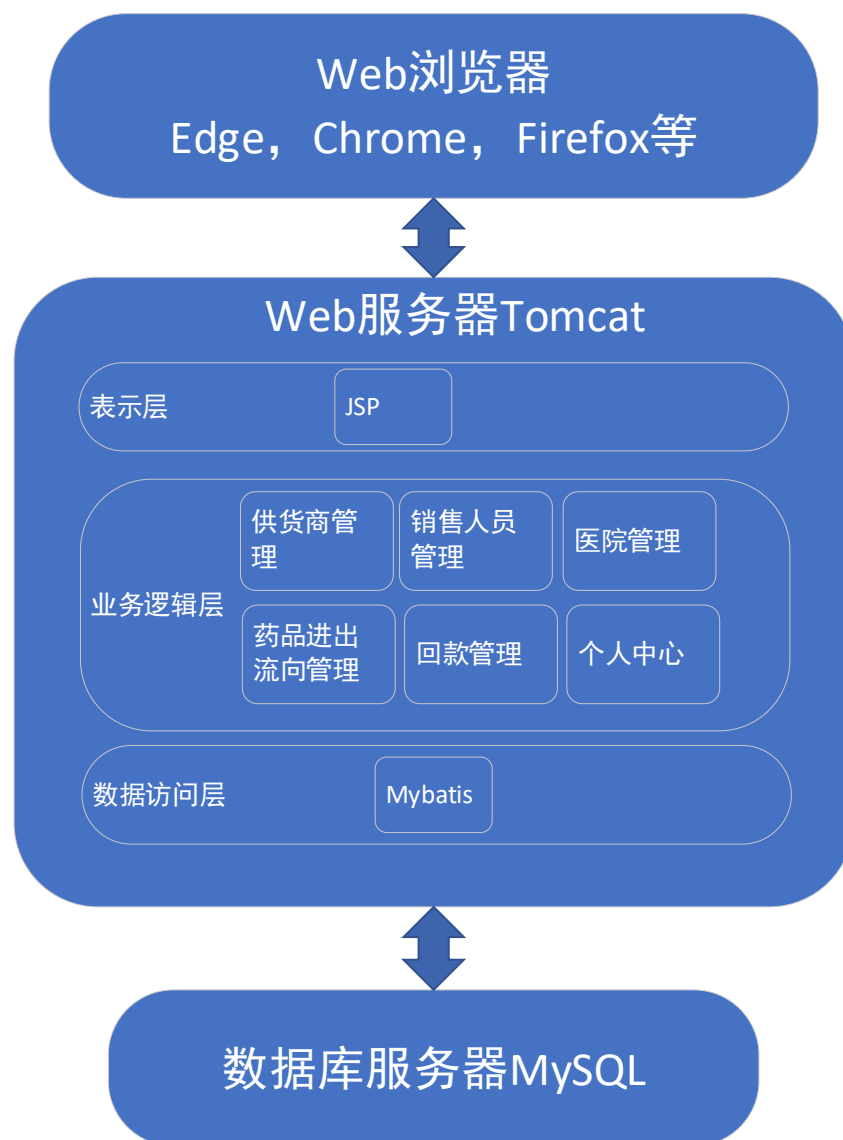


图 4.1 B/S 三层架构示意图

在 IE/chrome 浏览器层，系统与用户直接进行交互，用户通过它对系统进行操作及访问；系统依托 JSP 等进行界面设计，实现互动和显示功能。

整个系统架构中最重要的模块就是 Tomcat Web 服务器层。浏览器通过 Tomcat 实现系统的基本功能，Tomcat 被集成后统一设置在 Web 容器中。B/S 体系结构又进一步被细分为三层结构：一是 DAL（数据访问层），它的作用是对各项功能业务及逆行部署与设计；二是 BLL（业务逻辑层），负责实现各个功能模块的具体内容；三是 UI 层（表示层）。整个系统主要实现下列功能：供货商管理、销售人员管理、药品管理、医院管理、药品进出流向管理、回款管理、销售管理、个人中心等。在整个系统架构中，最底层的是 MySQL 数据库服务器，该层主要提供数据库的查询和存储等有关业务。

4.2 系统功能结构设计

系统功能结构图如图 4.2 所示。

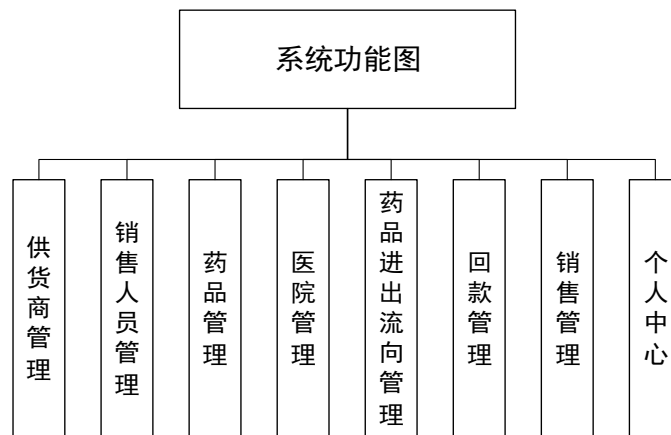


图 4.2 系统总体功能结构图

4.3 数据库结构设计

4.3.1 概念结构设计

数据库的概念设计整体属性 E-R 图如下图 4-3 所示：

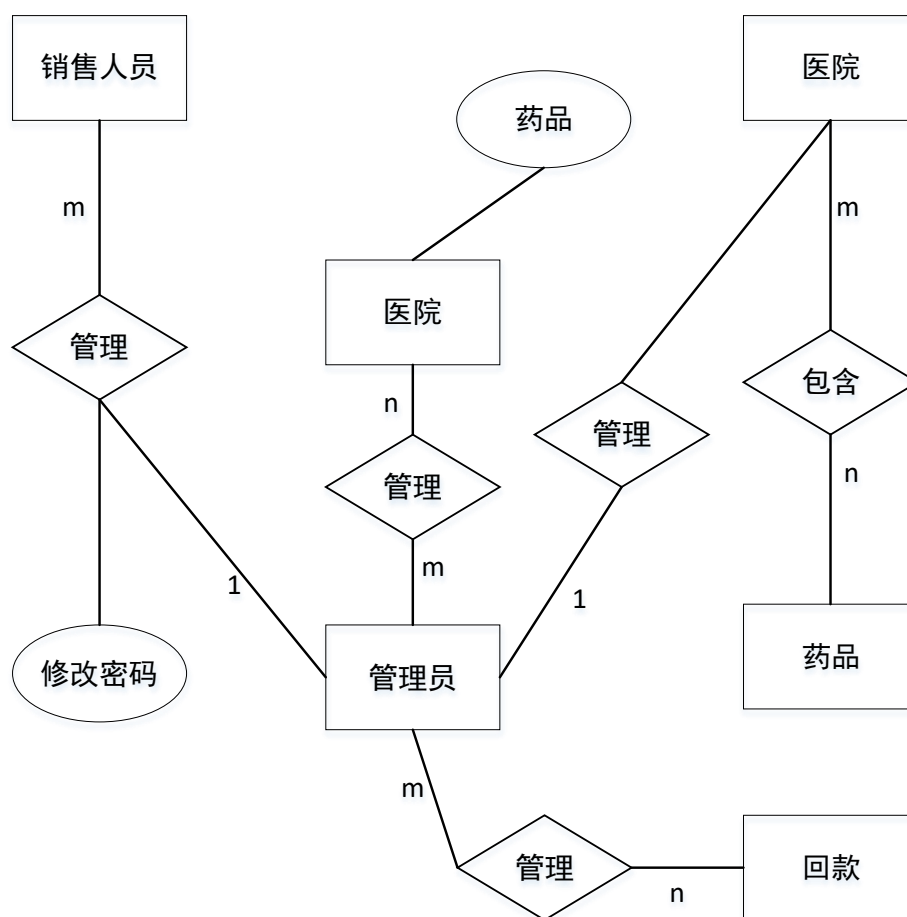


图 4.3 整体属性 E-R 图

然后根据具体的功能需求，对本系统的 E-R 图进行细化从而得到几种实体—关系模型，以下为部分实体的关系模型。

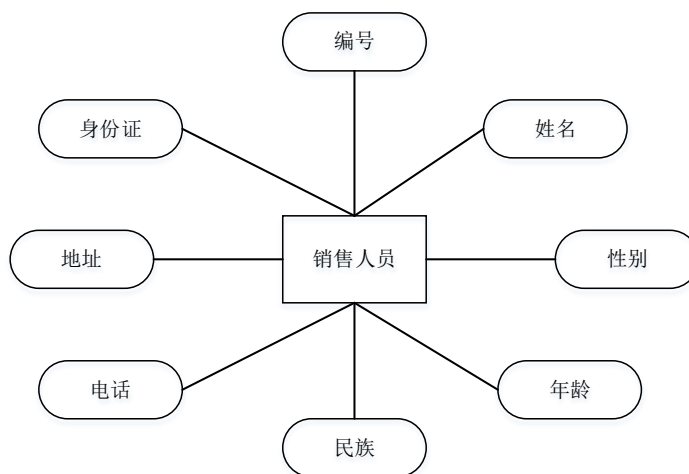


图 4.4 销售人员信息实体属性 E-R 分图

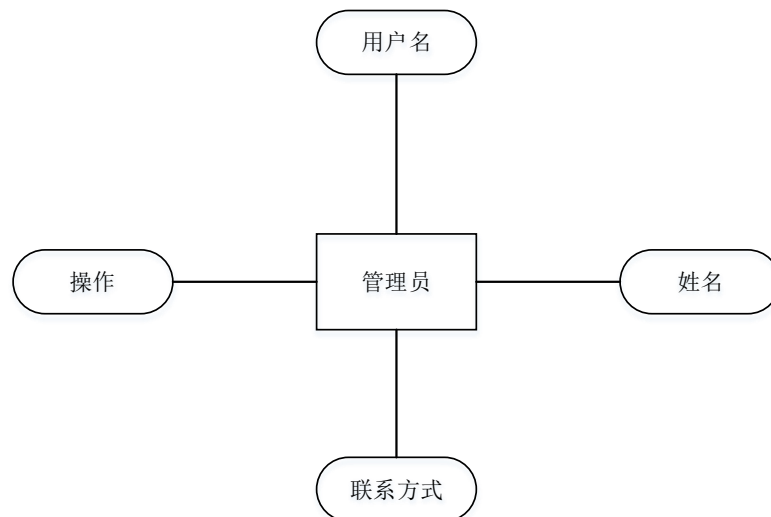


图 4.5 管理员信息属性 E-R 分图

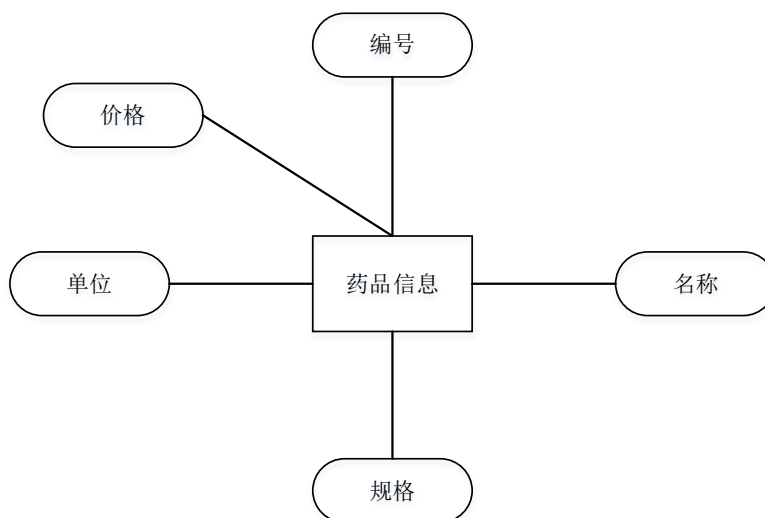


图 4.6 药品信息实体属性 E-R 分图

4.3.2 逻辑结构设计

将 E-R 图转化为关系模式实际上就是要将实体型、实体的属性和实体型之间的联系转化为关系模式。下面把图 4.3 的 E-R 图转换为关系模型。关系主码用下划线标出，外码用波浪线标出。

用户（用户编号，用户名，用户密码）

药品流水（流水编号，供货商，日期，药品，数量，销售员，医院，回款类型，是否回款）

供货商（供货商编号，姓名，地址）、

销售员（销售员编号，姓名，年龄，电话，身份证，住址）

药品（药品编号，名称，规格，单位，价格，库存）

医院（医院编号，名称，地址，类型）

以上是初步的逻辑结构设计，易看出这几个关系模式均满足 3 范式，可通过 power

designer 进行物理结构设计。

4.3.3 物理结构设计

通常关系数据库设计的主要内容包括关系模式选择存取方法，设计关系、索引等数据库文件得到物理存储结构。设计数据库表结构时需要遵循数据库设计原则，保证数据库的三阿达完整性：实体完整性，参照完整性，用户定义完整性。

本系统选用 MySQL 数据库管理数据，建立数据库名为 p7，以下是设计表的结构。

表 4.1 用户表（admin）表

字段名称	字段意义	字段类型	字段长度	是否主键	能否为空
adminid	编号	int	4	是	否
username	用户名	varchar	40	否	否
password	密码	varchar	40	否	否
realname	姓名	varchar	40	否	否
contact	联系方式	varchar	40	否	否

表 4.2 供应商（supplier）表

字段名称	字段意义	字段类型	字段长度	是否主键	能否为空
articleid	编号	int	4	是	否
name	名称	varchar	40	否	否
site	地址	varchar	40	否	否
principal	负责人	varchar	40	否	否
phone	电话	varchar	40	否	否

表 4.3 药品（drug）表

字段名称	字段意义	字段类型	字段长度	是否主键	能否为空
clazzid	编号	int	4	是	否
name	名称	varchar	40	否	否
standard	规格	varchar	40	否	否
unit	单位	varchar	40	否	否
price	价格	varchar	40	否	否
result	效果	varchar	40	否	否

表 4.4 药品进出流 (flow_direction) 表

字段名称	字段意义	字段类型	字段长度	是否主键	能否为空
courseid	编号	int	4	是	否
supplier	供货商	varchar	40	否	否
market	销售地点	varchar	40	否	否
drug	药品	varchar	40	否	否
hospital	医院	varchar	40	否	否
quantity	数量	varchar	40	否	否

表 4.5 销售 (sale) 表

字段名称	字段意义	字段类型	字段长度	是否主键	能否为空
name	姓名	varchar	40	是	否
age	年龄	varchar	40	否	否
phone	电话	varchar	40	否	否
card	身份证	varchar	40	否	否
address	住址	varchar	40	否	否

表 4.6 医院 (hospital) 表

字段名称	字段意义	字段类型	字段长度	是否主键	能否为空
name	姓名	varchar	40	是	否
addresses	住址	varchar	40	否	否
type	类型	varchar	40	否	否

4.4 系统用例图

管理员负责系统的整体管理，包括回款管理、医院管理、用户信息维护、供应商管理和药品信息管理。销售员负责向医院提供药品，并查询药品信息。供应商向医院提供药品，如图 4.7 所示。

- 系统的回款管理功能可以帮助医院及时收回药品款项，提高资金周转率。
- 系统的医院管理功能可以帮助医院管理患者信息、订单信息等。

- 系统的用户信息维护功能可以帮助医院管理用户信息，提高用户体验。
- 系统的供应商管理功能可以帮助医院管理供应商信息，提高供应链效率。
- 系统的药品信息管理功能可以帮助医院管理药品信息，提高药品管理效率

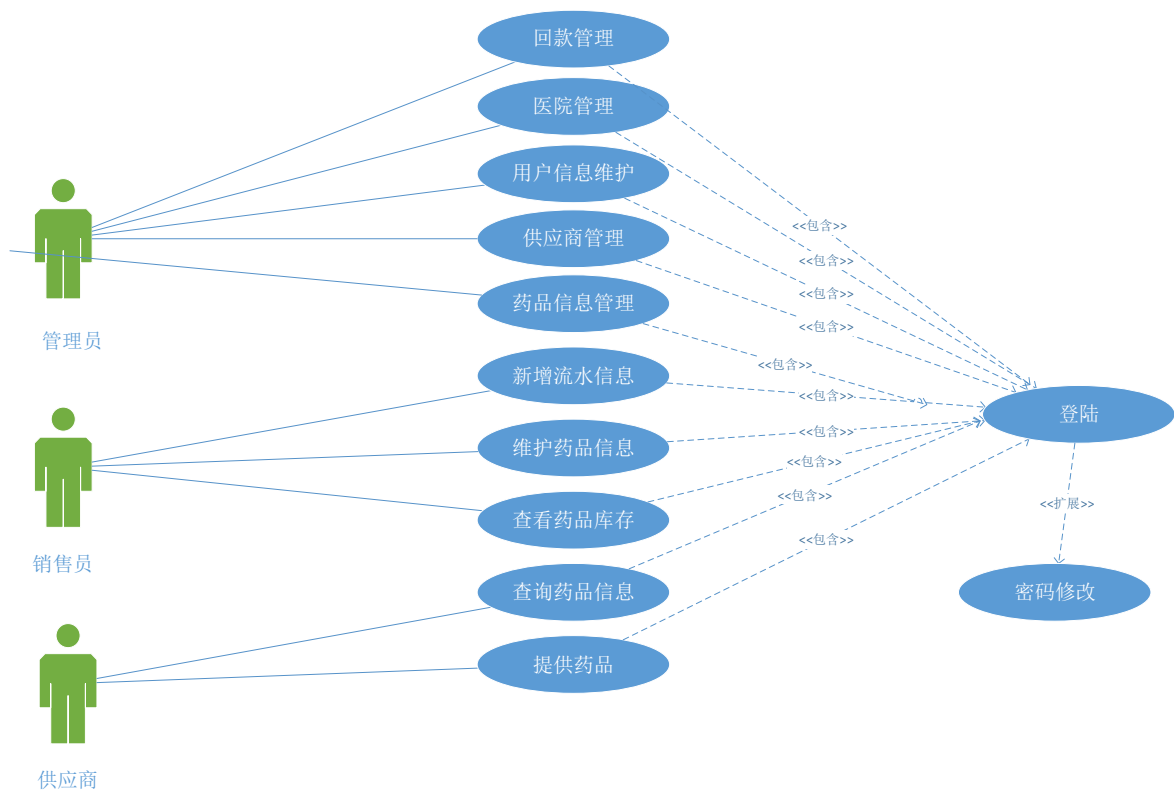


图 4.7 系统用例图

4.5 系统架构类图

展开包图，得到类图，它是静态结构图的架构，使各个种类之间的关系，表达了静态联系。系统类图如下图 4.8 所示。

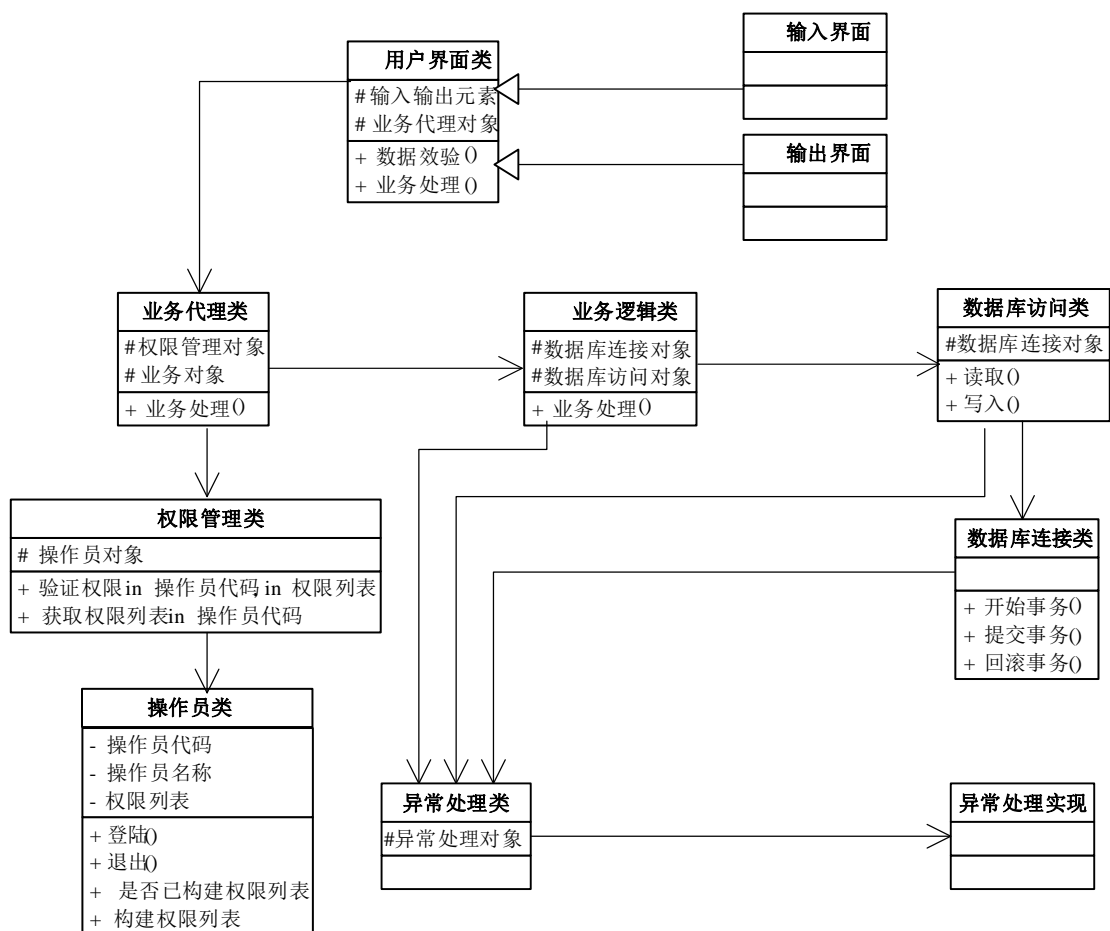


图 4.8 系统架构类图

4.6 事件跟踪图

系统架构类的工作流程：

1. 在用户输入请求后，用户界面对象接收到，并发送一个请求到服务代理对象。
2. 业务代理对象接收到请求之后，并向受权限管理对象发送的验证权限请求。
3. 权限管理对象把得到的验证权限的结果发给业务代理对象后。
4. 根据业务代理对象的验证结果进行以下处理：对权限的要求不符合的则返回消息;符合许可的请求时，该请求被转发到业务对象。
5. 业务对象进行业务处理。对于持久性的数据服务操作，操作时通过访问数据库访问对象，任何例外情况的处理都给异常处理对象进行处理。最后，把结果信息返回给业务代理对象。
6. 业务代理对象将处理信息返回给用户界面。

系统架构类的事件跟踪图如图 4.9 所示。

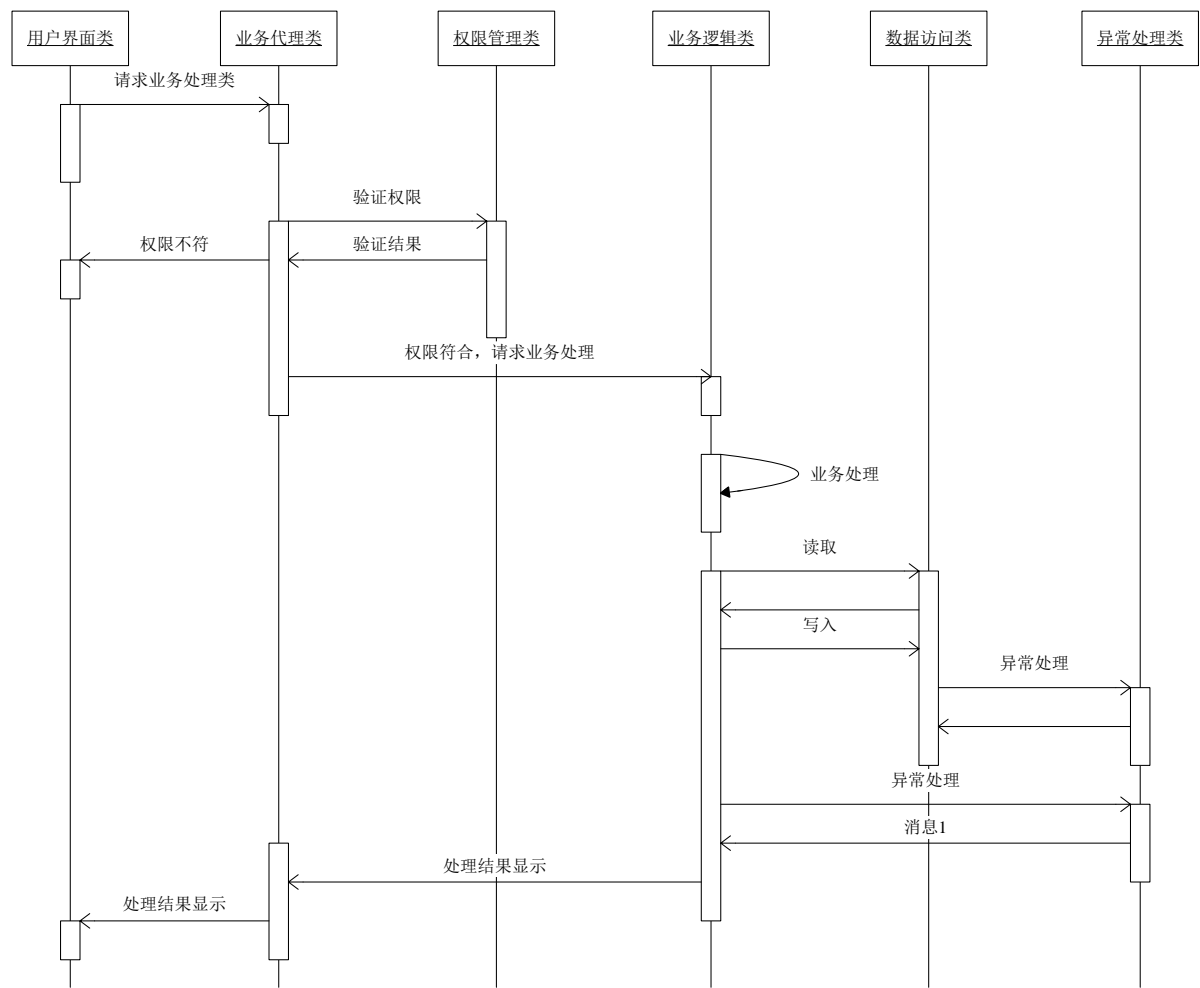


图 4.9 系统架构类的事件跟踪图

5 详细设计

5.1 系统配置

5.1.1 应用层配置

系统采用 Spring 框架进行开发。在 WEB-INF/web.xml[9]文件的配置代码如下：

```
<filter>
  <filter-name>Spring</filter-name>
  <filter-class>
    org.apache.Spring.dispatcher.FilterDispatcher
  </filter-class>
</filter>
```

```

        </filter-class>
    </filter>
    <filter-mapping>
        <filter-name>Spring</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
    <filter>
        <filter-name>Spring-cleanup</filter-name>
        <filter-class>
            org.apache.Spring.dispatcher.ActionContextCleanUp
        </filter-class>
    </filter>
    <filter-mapping>
        <filter-name>Spring-cleanup</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>

```

当加入这个配置到 web.xml 文件之后，在浏览器中敲入的访问系统的所有请求路径都会经过 openSessionInViewFilter 这个 filter 进行过滤。由这个过滤器进行转发相应的请求，进入 Spring 的工作流程中。这就是算是把 Spring 框架配置到程序中了。

5.1.2 数据库连接配置

服务器端程序采用 spring 框架来访问数据库：以下是 spring 连接数据库的配置文件，该配置是针对 dataSource 的连接数据库的配置：

```

<bean id="dataSource"
    class="org.apache.commons.dbcp.BasicDataSource">
    <property name="driverClassName"
        value="com.microsoft.jdbc.sqlserver.SQLServerDriver">
    </property>
    <property name="url"
        value="jdbc:sqlserver://localhost:1433;databaseName=training_db;user=sa">
    </property>
    <property name="username" value="sa"></property>
    <property name="password" value="1234"></property>
</bean>

```

5.1.3 SessionFactory 模式的配置

```

<bean id="sessionFactory"

```

```

class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
<property name="dataSource">
    <ref bean="dataSource" />
</property>
<property name="hibernateProperties">
    <props>
        <prop key="hibernate.dialect">
            org.hibernate.dialect.MySQLDialect
        </prop>
        <prop key="hibernate.show_sql">true</prop>
        <prop key="hibernate.hbm2ddl.auto">update</prop>
    </props>
</property>
<property name="mappingResources">
    <list>
        <value>com/model/Sysuser.hbm.xml</value>
        <value>com/model/Course.hbm.xml</value>
        <value>com/model/Check.hbm.xml</value>
        <value>com/model/Result.hbm.xml</value>
        <value>com/model/Job.hbm.xml</value>
    </list>
</property>
</bean>

```

6 系统实现

6.1 总体说明

系统实现称为编码或程序设计，主要按照系统设计说明书的要求，把系统中每一个功能模块用某种程序语言或某种开发工具予以实现，做出可以在计算机上执行的代码额过程。以下是药品管理系统的详细实现过程。

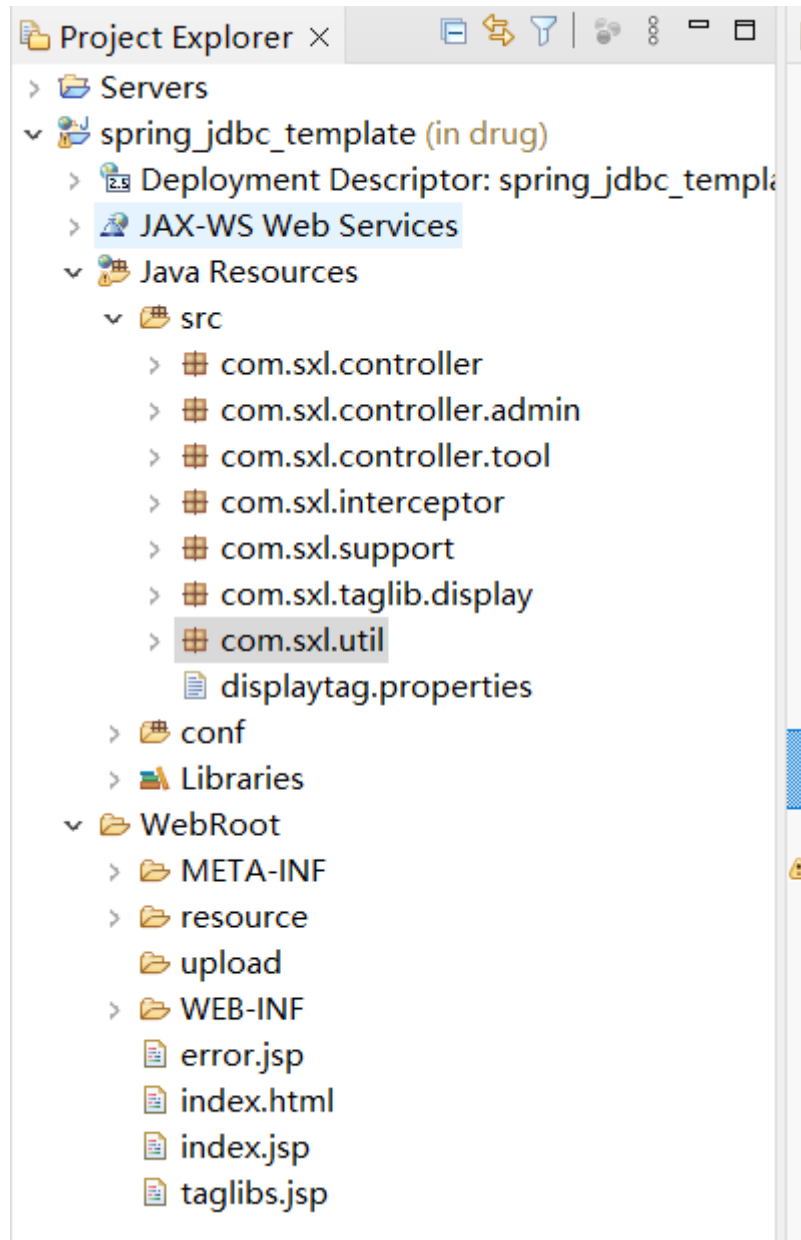


图 6.1 药品管理系统文件结构图

如图 6.1 所示,前端文件放在 webroot 目录底下,后的文件放在 Java Resources/src 下,主要分为 6 个后端 Java 包和一个交互接口 properties 文件。

Java Resources 模块包含以下子模块:

controller: 包含了控制器类,负责处理用户请求。

service: 包含了服务类,负责处理业务逻辑。

dao: 包含了数据访问对象 (DAO),负责访问数据库。

model: 包含了数据模型,用于存储数据。

util: 包含了工具类,用于提供通用的功能。

Servers 目录下包含了该项目部署的服务器配置信息。

WebRoot 目录下包含了以下子目录：

- META-INF 目录包含了该项目的元数据文件，例如 MANIFEST.MF 文件。
- resource 目录包含了该项目的静态资源文件，例如 CSS、JavaScript 和图片文件。
- upload 目录包含了该项目的文件上传目录。
- WEB-INF 目录包含了该项目的 Web 应用程序的内部资源，例如 web.xml 文件和类库。

6.2 部分核心代码

6.2.1 前端主界面 main.JSP

```
<div id="header">
  <div class="logo">药品管理系统</div>
  <div class="navigation">
    <ul>
      <li>欢迎您! </li>
      <li><a href="">${adminBean.username }</a></li>
      <%--
      <li><a href="">修改密码</a></li>
      <li><a href="">设置</a></li> --%>
      <li><a href="${ctx }/adminLogin/out.html">退出</a></li>
    </ul>
  </div>
</div>
<div id="content">
  <div class="left_menu">
    <ul id="nav_dot">
      <li>
        <h4 class="M1"><span></span>个人中心</h4>
        <div class="list-item none">
          <a href='${ctx }/admin/password.html' target="page">修改密码</a>
        </div>
      </li>
      <li>
        <h4 class="M1"><span></span>供货商管理</h4>
        <div class="list-item none">
```

```

        <a href='${ctx}/admin/ghs/frame.html?flag=1' target="page">供货商列表
    </a>

    </div>
</li>
<li>
    <h4 class="M1"><span></span>销售人员管理</h4>
    <div class="list-item none">
        <a href='${ctx}/admin/xs/frame.html?flag=1' target="page">销售人员列
表</a>

    </div>
</li>
<li>
    <h4 class="M1"><span></span>药品管理</h4>
    <div class="list-item none">
        <a href='${ctx}/admin/yp/frame.html?flag=1' target="page">药品列表
    </a>

    </div>
</li>
<li>
    <h4 class="M1"><span></span>医院管理</h4>
    <div class="list-item none">
        <a href='${ctx}/admin/yy/frame.html?flag=1' target="page">医院列表
    </a>

    </div>
</li>
<li>
    <h4 class="M1"><span></span>药品进出流向管理</h4>
    <div class="list-item none">
        <a href='${ctx}/admin/flow/frame.html?flag=1' target="page">药品进出
流向列表</a>

        <a href='${ctx}/admin/yp/frame2.html' target="page">库存管理</a>
    </div>
</li>
<li>
    <h4 class="M1"><span></span>回款管理</h4>
    <div class="list-item none">
        <a href='${ctx}/admin/flow/frame2.html?flag=1' target="page">回款药品
流向列表</a>

        <a href='${ctx}/admin/tj/tj1.html' target="page">月回款金额统计</a>
    </div>
</li>
<li>
    <h4 class="M1"><span></span>销售管理</h4>
    <div class="list-item none">

```

```

        <a href='${ctx}/admin/all/frame.html?flag=1' target="page">医院流向查
    询</a>
        <a href='${ctx}/admin/all/frame.html?flag=2' target="page">药品流向查
    询</a>
        <a href='${ctx}/admin/all/frame.html?flag=3' target="page">销售人员销
    售情况查询</a>
        <a href='${ctx}/admin/all/list4.html' target="page">销售日报表</a>
        <a href='${ctx}/admin/all/list5.html' target="page">销售月报表</a>
    </div>
</li>
</ul>
</div>
<div class="m-right">
    <div class="main">
        <iframe src='${ctx }/admin/main.html' width="100%" height="100%"
    frameborder="0" scrolling="yes" name="page"></iframe>
    </div>
</div>
</div>

```

该 JSP 文件是药品管理系统的首页，主要功能是提供导航菜单和主页面。核心部分是导航菜单，它位于页面左侧。导航菜单包含了系统的所有功能模块，用户可以通过点击菜单项来访问相应的功能。导航菜单采用了树形结构，可以方便用户查找和使用功能。

6.2.2 控制类部分核心代码

以下是控制器类的核心部分：

首先获取用户信息（admin），从会话 Session 中获取名为 "adminBean" 的属性，并将其作为 Map 对象返回。该属性通常存储了当前登录管理员的信息。

```

public Map getAdmin(HttpServletRequest request) {
    Map customerBean = (Map) request.getSession().getAttribute("adminBean");
    return customerBean;
}

```

然后构造 JSON 响应，使用 JacksonJsonUtil 将对象转换为 JSON 字符串，并设置 HTTP 状态码为 200（OK）。

```

public ResponseEntity<String> renderMsg(Boolean status, String msg) {
    // ... 构造 JSON 字符串
    return new ResponseEntity<String>(sb.toString(), initHttpHeaders(), HttpStatus.OK);
}

public ResponseEntity<String> renderData(Boolean status, String msg, Object obj) {
    // ... 构造 JSON 字符串包含数据
    return new ResponseEntity<String>(sb.toString(), initHttpHeaders(), HttpStatus.OK);
}

```



```
}
```

接着生成下拉框数据的 JSON 响应，生成用于填充下拉框的 JSON 数据，满足 AJAX 请求。

```
public ResponseEntity<String> renderComboBoxAjax(Object obj) {  
    return new ResponseEntity<String>(JacksonJsonUtil.toJson(obj), initHttpHeaders(),  
    HttpStatus.OK);  
}
```

最后是数据库的访问，注入了 DBHelper 对象，用于简化数据库操作，封装常用的数据库操作方法。

```
@Autowired  
public DBHelper db;
```

以上是一个基本控制类的 Java 代码，其其它控制类模块基本流程与此类似，篇幅有限，不做过多讲解。

6.3 功能模块实现

6.3.1 系统登陆实现

系统登录模块实现用户登录系统实现管理功能，用户需要输入用户名、密码以及选择用户类型进行登录。系统登录实现界面如图 6.2 所示。



图 6.2 登陆界面

6.3.2 系统功能设计实现

系统主要实现了供货商管理、销售人员管理、药品管理、医院管理、药品进出流向管理、回款管理、销售管理、个人中心等功能模块。

(1) 供货商管理

供货商管理模块主要实现对供货商信息的查询、删除、修改、添加等功能。供货商管理模块实现界面如图 6.3 所示。

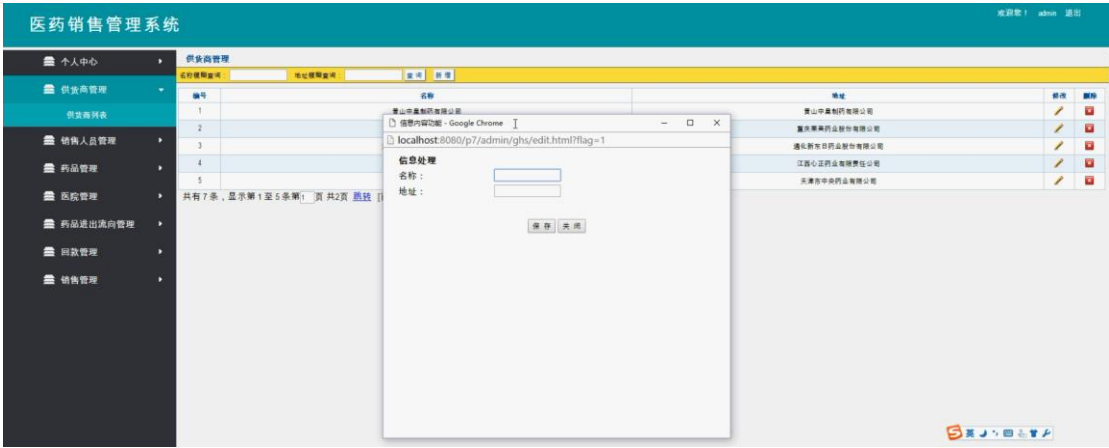


图 6.3 供货商管理模块实现界面

(2) 销售人员管理

销售人员管理模块主要实现管理员对销售人员信息的查询、删除、修改、添加等功能。销售人员管理模块实现界面如图 6.4 所示。

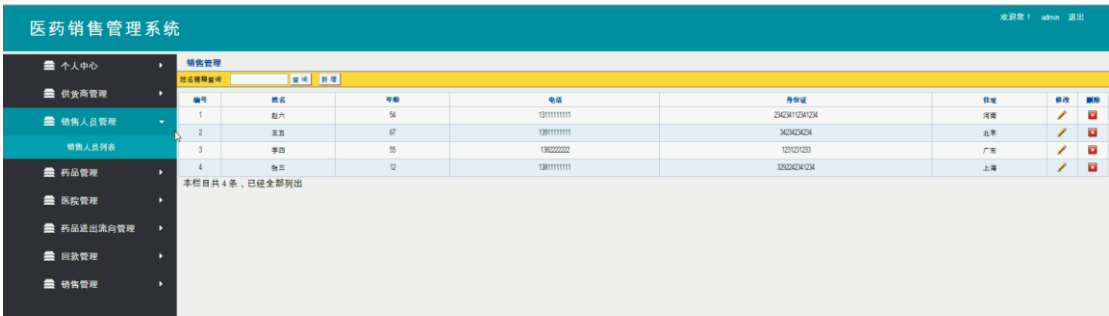


图 6.4 销售人员管理模块实现界面

(3) 药品管理

药品管理模块主要实现了管理员对药品信息的查询、添加、修改、删除等功能。药品管理模块实现界面如图 6.5 所示。



图 6.5 药品管理模块实现界面

(4) 医院管理

医院管理模块主要实现管理员对医院信息的查询、删除、修改、添加等功能。医院管理模块实现界面如图 6.6 所示。

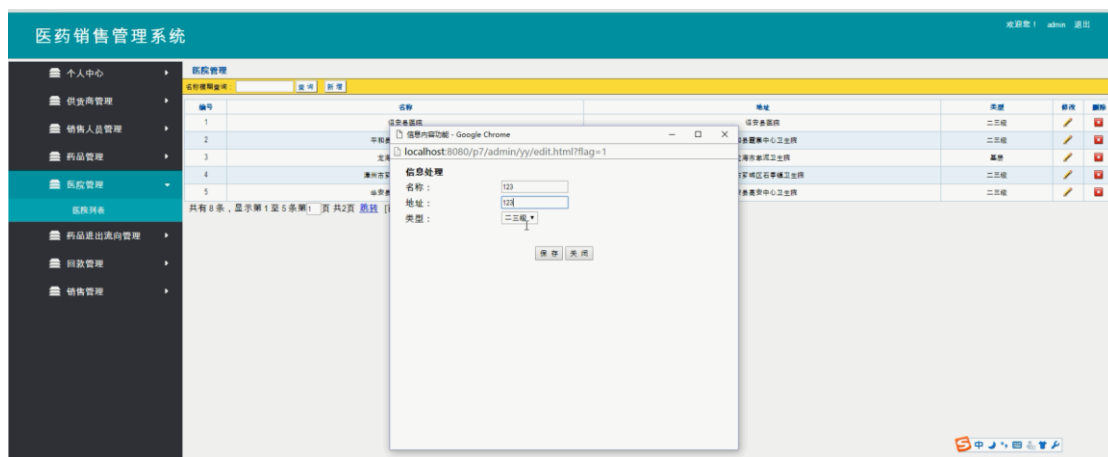


图 6.6 医院管理实现界面

7 系统测试

7.1 测试方法

本系统测试所用方法包括以下两个：

- 1) 黑盒测试（功能测试）：将整个系统当成一个整体，在不了解系统内部护具结构的基础上，通过测试来检测各个功能是否能实现目标需求，从而达到检验系统外部结构的作用。黑盒测试能后检测系统是否能够做到预先额功能需求以及合计需求。
- 2) 白盒测试（结构测试）[10]：一种依赖系统内部结构和逻辑处理关系的猜测是方法。在充分了解系统内部结构逻辑结构的基础上，检查系统各项功能和性能是否符合设计目标。

7.2 测试的主要内容

测试过程分为哪会保证测试，即质量几个阶段：代码审查，单元测试，集成测试，确认测试和系统测试。

(1) 单元测试

集中在一个小软件单元设计审查 - 模块是一个测试单元，意识和定义模块的功能实际发现，这种情况不符合测试说明和编码错误。

(2) 集成测试

集成测试模块安装在根据进行集成测试的设计要求，其主要目的是为了发现问题相关的接口。例如有可能是一个问题，由于疏忽，与其他模块的模块，并造成有害影响；不能产生所需的功能键，组合功能;看起来可以接受的个别错误是不可接受的，当积累；可能有错误，如整个数据结构。

(3) 确认测试

该试验的目的是验证，该系统能够工作如期，根据需要，以显示未来的用户。集成测试后，已经把一个完整的软件系统，设计按照完成安装的所有模块，从根本上推动问题接口错误的，那么软件的有效性将能够得到进一步的确认，试验工作证实，它是功能和软件的性能，如用户所期望的。

(4) 测试系统

在软件开发完成后，并且系统将最终在包的其它部分，在测试系统上执行。
单独的测试系统主要从以下几个方面：
对后台登录功能进行测试，后台登录界面如下图 7.1 所示：



图 7.1 登录模块测试界面

可见，只有当用户名和密码都输入正确时才会登录成功。接下来对添加药品进行测试，添加药品操作界面如下图 7.2 所示：

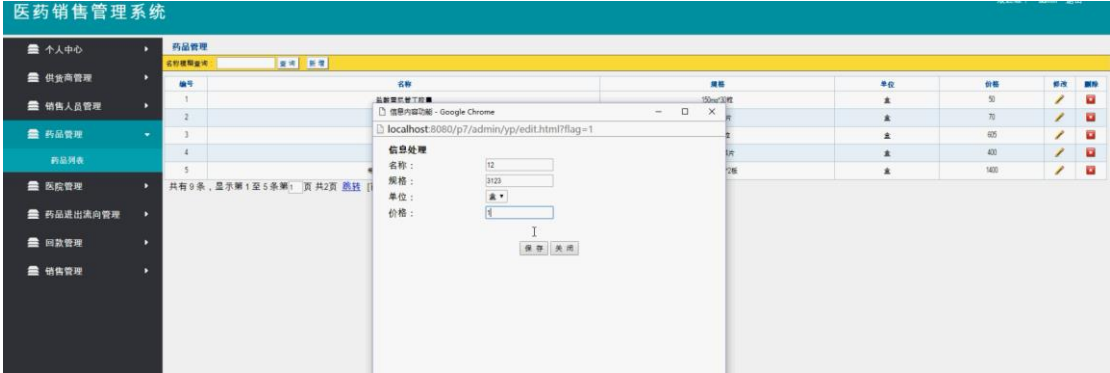


图 7.2 添加药品测试页面

添加药品成功界面如图 7.3 所示：

药品管理						
名称模糊查询						
编号	名称	规格	单位	价格	修改	删除
1	12	3123	盒	12		
2	盐酸雷尼替丁胶囊	150mg*30粒	盒	50		
3	盐酸环丙沙星片	0.25g*12片	盒	70		
4	抗宫炎胶囊	0.5g*72粒	盒	605		
5	抗宫炎片	0.35g*144片	盒	400		

共有 10 条，显示第 1 至 5 条第 1 页 共 2 页 跳转 (首页/上一页) 1, 2 (下一页/末页)

图 7.3 添加药物测试界面

删除该药物，测试成功如图 7.4。

localhost:8080 显示
确定要删除吗?

确定 取消

药品管理						
名称模糊查询						
名称	规格	单位	价格	修改	删除	
12	3123	盒	12			

图 7.4 删除药物测试界面

对药物进行查询，测试成功，结果如图 7.5 所示。

药品管理		
名称模糊查询: 抗宫炎胶囊 查询 新增		
编号	名称	
1	抗宫炎胶囊	

共 1 条，已经列出

图 7.5 查询药物测试界面

对药物进行修改，测试结果正常，结果如图 7.6 所示。

药品管理

名称模糊查询: 抗宫炎胶囊 查询 新增

编号	名称	规格	单位	价格
1	12	3123	盒	12
2	盐酸雷尼替丁胶囊	150mg*30粒	盒	50
3	盐酸环丙沙星片	0.25g*12片	盒	70
4	抗宫炎胶囊	0.5g*72粒	盒	605
5	抗宫炎片	0.35g*144片	盒	400

共有 9 条，显示第 1 至 5 条第 1 页 共 2 页 跳转 (首页/上一页) 1, 2 (下一页/末页)

信息处理

名称: 盐酸雷尼替丁胶囊

规格: 150mg*30粒

单位: 盒

价格: 50

保存 关闭

信息处理

名称: 成功

规格:

单位:

价格:

确定

保存 关闭

图 7.6 修改药物测试界面

对其他模块的功能测试功能正常，篇幅有限，在此不再展开。

8 回顾与展望

8.1 回顾总结

本文主要阐述了如何利用 Spring + SpringMVC + MyBatis 框架开发一个性能优化、可扩展性强和安全可靠的药店管理管理系统。系统主要实现了供货商管理、销售人员管理、药品管理、医院管理、药品进出流向管理、回款管理、销售管理、个人中心等功能模块。然后根据这些功能进行分析设计，对所有到的技术 Spring、SpringMVC 等进行简要的介绍。接着进行需求分析的描述。接着是概要设计和详细设计。描述这个系统的部分功能的实现方案及过程。

本设计所实现的是一个药店管理管理系统，主要介绍了 JSP、Spring、SpringMVC 技术。系统按照总体设计、数据库设计、各个模块设计和代码分析，药品管理管理系统的基本功能已得到实现。由于时间及本人所学有限，该系统还有许多地方需要改进，并且还有许多自己未起到的功能，本人会在以后的学习过程中进一步加强和完善。

8.2 优点与不足之处

本系统具有以下优点：

1. 该系统可以运行在多个操作系统平台（windows10、Windows11 等），数据库采用 MySQL，开发语言选择 Java，可移植性好。
2. 该系统界面简单，操作方便，容易使用。

但也存在以下缺点：

1. 界面跳转复杂，接口不能被刷新，可以改进。
2. 功能比较简单，没有进一步提高一些选修的程序等，不能更好的为用户提供服务。
3. 数据库设计有冗余，需要进一步优化。

8.3 展望

由于理论与系统实现过程时间有限，另外设计实验的外在条件有限，本系统还有待进一步改进与完善。有一些挺不错的想法我认为后续有时间可以完善一下。

- 1) 系统的数据库部分可以以租用云服务器器的形式（尝试过但是失败了），将项目代码打包上线，降低成本，减少后续维护工作。
- 2) 在安全过滤防护措施方面，系统除了具有针对性的展开防御，有效防御脚本文件展开的攻击，还需要充分重视其他常见的脚本攻击，如跨站 DOS/DDOS 攻击等。并且对用户数据进行加密，确保系统更加可靠安全。
- 3) 可以使用机器学习算法对药品功效进行等一些基本操作进行模型训练，实现一些自然语言处理的任务（后或者直接外接一个 GPT 接口直接智能化处理用户请求），辅助管理员的维护。

参考文献

- [1] Yan Z , Shi Z , Wu Y ,et al.Wireless, noninvasive therapeutic drug monitoring system for saliva measurement toward medication management of schizophrenia[J].Biosensors & Bioelectronics: The International Journal for the Professional Involved with Research, Technology and Applications of Biosensors and Related Devices, 2023.
- [2] <https://blog.mybatis.org/>
- [3] <https://spring.io/>
- [4] 骆鑫.医院信息化药品管理系统的设计与实现[D].电子科技大学[2024-01-01].DOI:CN KI:CDMD:2.1013.329130.
- [5] 张振. 新医大肿瘤医院药品管理系统的设计与实现. (Doctoral dissertation, 大连理工大学).
- [6] 刘嘉富,李研琼.基于 Web 的药品管理系统的设计与实现[J].电脑知识与技术:学术版, 2021.
- [7] 甄海涛, 杨卓林, and 赵寒涛. "基于 SSM 架构的项目管理平台设计." 自动化技术与应用 38.5(2019):3.
- [8] [简述 Spring MVC 的请求处理流程以及整体框架结构_springmvc 请求流程-CSDN 博客](#)
- [9] <https://github.com/tys1998/DIMS?tab=readme-ov-file#%E5%BA%94%E7%94%A8%E7%B3%BB%E7%BB%9F%E8%AE%BE%E8%AE%A1>
- [10] 杨东. "基于 Spring Boot 的学生信息管理系统的设计与实现."
- [11] 药品数据来源[医药招商_药品招商_医药代理专业医药网站_药源网 \(yaopinnet.com\)](#)

致 谢

感谢明老师一个学期以来的教导，学生朽木。